DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

Balance-Enforced Multi-Level Algorithm for Multi-Criteria Graph Partitioning

Rémi BARAT [1,2]
Cédric CHEVALIER [1]
François PELLEGRINI[2,3]
Speaker: Jean-Sylvain CAMIER [1]
[1] CEA, DAM, DIF, F-91297 Arpajon, France
[2] University of Bordeaux, France
[3] INRIA, France

28 February 2017
SIAM Conference on Computational Science and Engineering

Outline

Outline

Context

In the field of numerical simulation, one challenge is to reduce the restitution time.

We consider simulations:

- Running on distributed memory architectures
- Using 2D or 3D meshes
- Coupling several physics (multi-physics simulations)

Running these simulations efficiently needs to:

1. Balance the workloads of each processor for each phase of the computation
2. Overlap or minimize communications
3. Take care of memory accesses
4. Exploit full processor characteristics

We focus on the first 2 items.

Example

Algorithm 1 Example of the temporal loop of a multi-physics simulation
(2 computing phases: f and g, coupling the variables X and Y)

for $t \in [0, t_{end} - 1]$ **do**

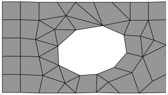
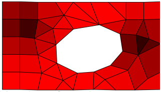
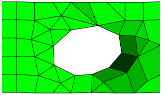
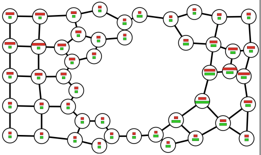
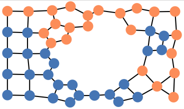
$X(t+1) \leftarrow f(t+1, X(t), Y(t))$ # First computing phase

$Y(t+1) \leftarrow g(t+1, X(t+1), Y(t))$ # Second computing phase

end for

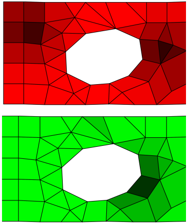
Objectives:

1. Balance the workloads of each processor *for each phase*
2. Minimize communications between processors

Input		Model	Output
Geometry	Weights	Multi-criteria Dual Graph (1 color = 1 criterion)	Graph/Mesh Partition (1 color = 1 part)
	→ Weights for phase 1:  → Weights for phase 2: 		 imbalance c1: 1.9% imbalance c2: 2.9% edgecut: 11

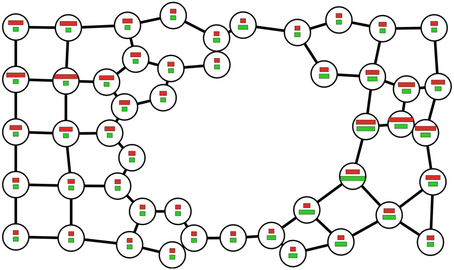
Model

Mesh



$\rightarrow$

Dual Graph $G = (V, E)$



Problem : Graph partitioning

We search for a partition P of V respecting:

1. some **constraints**: load-balancing
   (imbalance of P is for all criteria smaller than a given tolerance)
2. an **objective**: minimize the edgecut
   (sum of the weights of the edges cut by the partition).

NP-Hard Problem.

Main existing software:

Software	Representations	Multi-criteria	Origin	Limitations for us
Scotch	Topological	No	INRIA	No multi-criteria (yet)
MeTiS	Topological	Yes	University of Minnesota	Does not often meet the balance constraints
PaToH	Topological	Yes	Bilkent University	No parallel version No open-source version Does not always meet the balance constraints
Zoltan	Geometric Topological	Yes No	Sandia National Laboratories	Geometric representations are inefficient

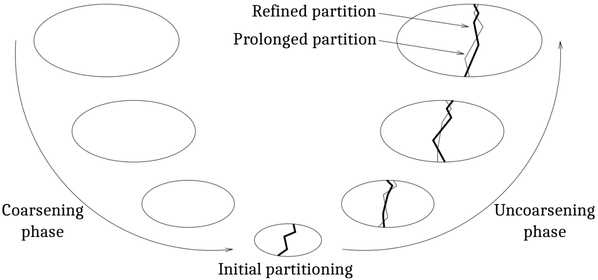
$\Rightarrow$ Lack of efficient multi-criteria partitioning tools.

Outline

Classic algorithm: The multi-level framework

A 3-phases algorithm:

1. Coarsening
2. Initial partitioning of the coarsened hypergraph
3. Uncoarsening and refinement

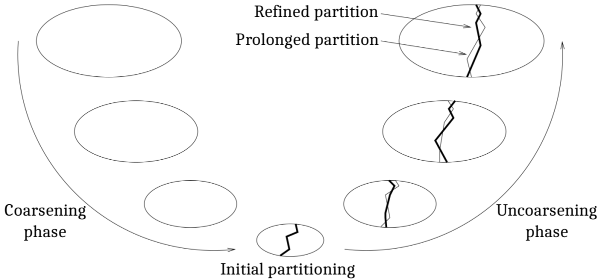


Our approach: Multi-level multi-criteria algorithm

A 3-phases algorithm:

1. Coarsening
   → (Ongoing) Selection of a matching policy improving the quality of the returned solutions
2. **Initial partitioning of the coarsened hypergraph**
   → New algorithm focusing on balance constraints
3. **Uncoarsening and refinement**
   → Adapted Fiduccia-Mattheyses algorithm

$\left.\begin{matrix} 2 \\ 3 \end{matrix}\right\}$ Implemented in Scotch



Scotch multi-criteria
Currently implemented algorithm

- Coarsening:
  - Classical algorithm: Heavy-Edge-Matching
- Initial Partitioning:
  - Local optimization of the balance
  - Enforce balance for all the constraints
  - Presented in SIAM CSC 2016
- Refinement:
  - Fiduccia-Mattheyses algorithm: local optimization of the edgecut with hill-climbing
  - Moves that unbalance the partition more than the tolerance are forbidden

$\Rightarrow$ Finding a balanced partition of the coarsest graph guarantees that the returned partition will respect the balance constraints.

Outline

Instances and Parameters

Comparison between

- MeTiS (5.1.0)
- PaToH (3.2)
- Scotch (6.1.0 - not yet released multi-criteria version)

Instances considered:

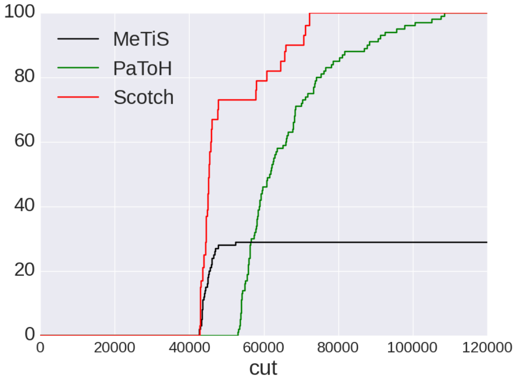
Name	Geometry	Vertices	Edges	Criteria	Edge weights
mushroom_1	2D	22800	45253	3	$\sqrt{w_i + w_j}$
mushroom_2	2D	22800	45253	3	independant
onera	3D	85567	166817	3	$\sqrt{w_i + w_j}$
wave	3D	156317	1059331	3	$\sqrt{w_i + w_j}$

Partitioning parameters:

- Imbalance tolerance 5%
- Number of parts: 2
- Each algorithm is executed 100 times on each instance with random seeds.

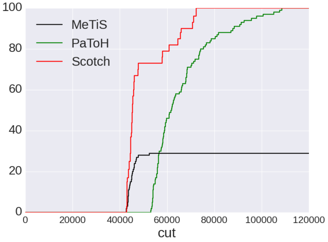
$f_{algo}(x) =$ number of valid solutions of $algo$ of cut $\leq x$.

- We want to minimize the cut of each returned solution: left is better

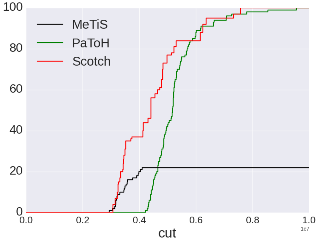


mushroom_1

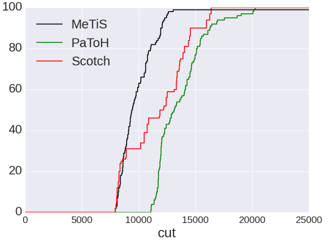
Results
2 - Comparison on the quality of the returned solutions



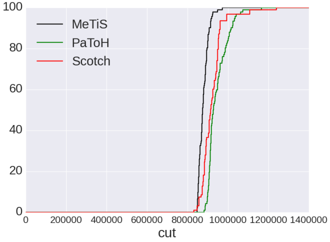
mushroom_1



mushroom_2



onera



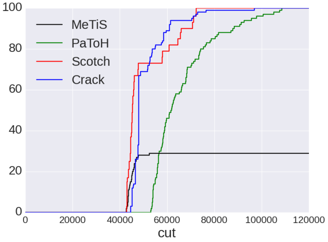
wave

Outline

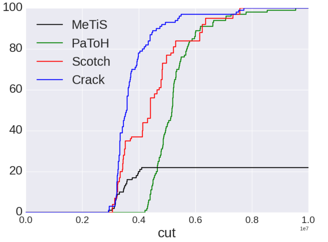
Improving robustness

- Python prototype: Crack
- Allows to test different parameters on the multi-level
- An example: settings of the coarsening parameters
  - Heavy-Edge-Matching
  - Particular order of the vertices:
    sorted by increasing degrees
  - Do not allow to match 2 vertices if the resulting vertex is overweighted

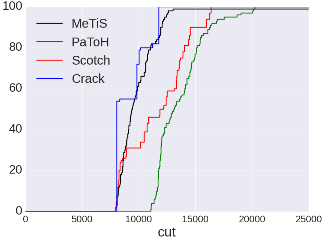
Comparison of Crack on the previous instances



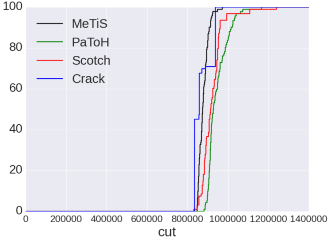
mushroom_1



mushroom_2



onera



wave

Outline

Conclusion

Summary:

- Scotch next release will support multi-criteria graph partitioning
- This new version implements a new initial partitioning and sets the refinement phase to guaranty that the returned solution will respect the balance constraints
- Scotch results are promising

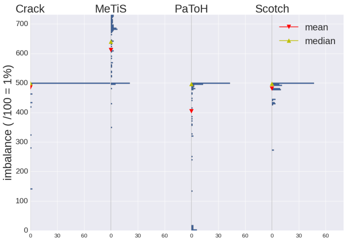
Perspectives:

- Enforce the algorithm robustness by the study of variations of the algorithms
- Validation on more instances
  Validation on a simulation code
- Set up of a parallel version of the algorithms

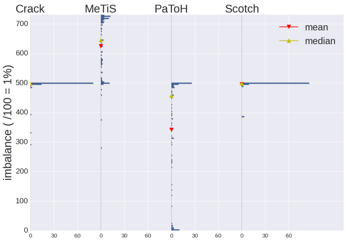
Thank you

Commissariat à l'énergie atomique et aux énergies alternatives
Centre DAM Île-de-France | F-91297 Arpajon
T. +33 (0)1 69 26 40 00

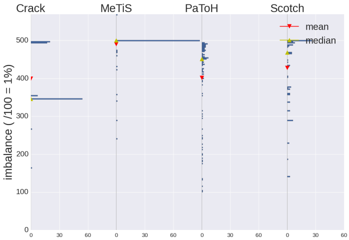
Imbalance distributions



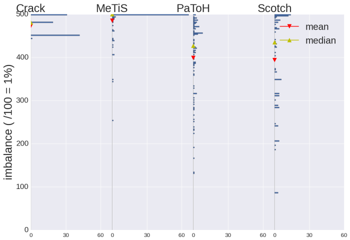
mushroom_1



mushroom_2



onera



wave